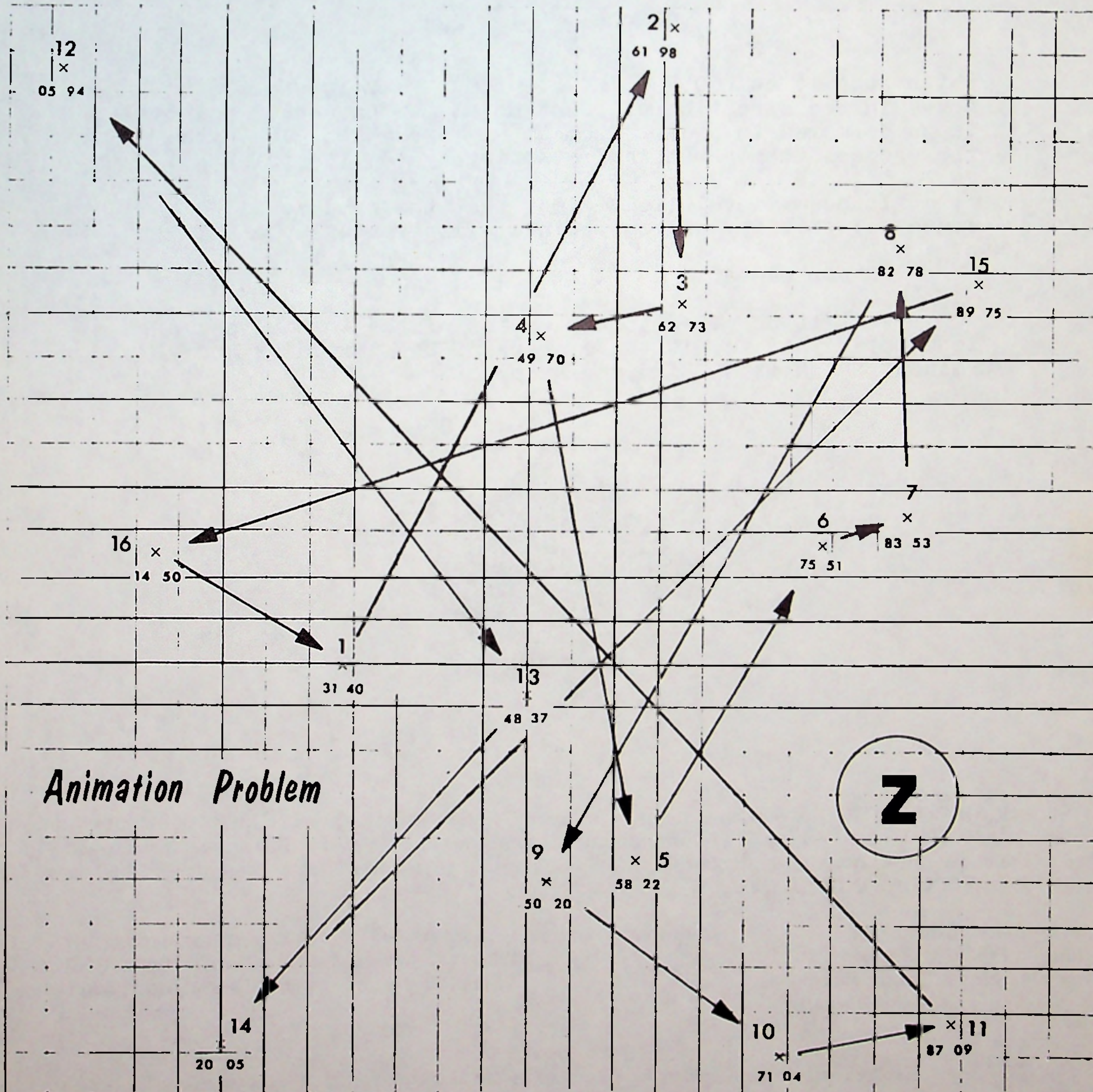
47

$2.50

Popular Computing

February 1977 Volume 5 Number 2



Animation Problem

Sixteen points are located on a 100 x 100 grid, as shown on the cover. The points are to move as follows:

Each point moves 1/7 of the way toward the next higher numbered point. Thus, for example, point No. 5 should move to a new position given by:

$$x = 58 + \frac{1}{7}(75 - 58)$$

$$y = 22 + \frac{1}{7}(51 - 22)$$

which puts it at (60.42857, 26.14286). Point No. 16 moves in the direction of point No. 1. The set of points is to be moved in order, from No. 1 through No. 16, with the process then repeating, starting again with point No. 1.

It seems intuitively clear that the process will converge; that is, all the points will eventually meet.

If so, where?

Will it be close to the center of the initial set of 16 points; that is, at the average of the initial x and y values, which is 55.3125 and 48.6875?

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $20.50 per year, or $17.50 if remittance accompanies the order. For Canada and Mexico, add $1.50 per year to the above rates. For all other countries, add $3.50 per year to the above rates. Back issues $2.50 each.

Knuth 2

--by Donald E. Knuth Computer Science Department Stanford University Stanford, California 94305

An Approach to Floyd's Problem

Last month I discussed some of the virtues of Floyd's problem, which is to divide the numbers

$$\sqrt{1}, \sqrt{2}, \ldots, \sqrt{50}$$

into two parts whose sums are as equal as you can make them after ten seconds of computer time. Here is the way I decided to approach the subject; perhaps some reader will have a much better idea.

Since $\frac{1}{2}(\sqrt{1}+\sqrt{2}+\ldots+\sqrt{50})$ is equal to

$$119.51790\ 03017\ 60392\ 24702\ 02231\ldots,$$

we seek the subset of $\{\sqrt{1}, \sqrt{2}, \ldots, \sqrt{50}\}$ whose sum least exceeds this value.

In the first place it is helpful to estimate the kind of results that might be expected. Most subsets of the given set have about 25 elements, and in fact the number of subsets with exactly 25 elements is 50!/25!25!; this is about

$$\frac{2^{50}}{\sqrt{25\pi}},$$

according to Stirling's approximation. [The exact number is 12641 06064 37752, compared to

$$2^{50} = 1\ 12589\ 99068\ 42624.]$$

All of these subsets have a sum which lies between

$$\sqrt{1}+\sqrt{2}+\ldots+\sqrt{25} \approx 85.6 \quad \text{and} \quad \sqrt{26}+\sqrt{27}+\ldots+\sqrt{50} \approx 153.4,$$

and they tend to cluster about the average value 119.5. So we have more than 10^{14} numbers packed into an interval of length less than 70; therefore $70/10^{14}$ appears to be a conservative estimate for the amount by which the best partitioning exceeds

$$\frac{1}{2}(\sqrt{1}+\sqrt{2}+\ldots+\sqrt{50}).$$

But not all of these subsets will give different sums. In the first place, the seven numbers

$$\sqrt{1}, \sqrt{4}, \ \ldots\ , \sqrt{49} = 1,2,\ldots,7$$

will yield only integer values (and it is easy to verify that each integer from 0 to 28 can be represented). Our problem therefore reduces to finding a subset of the 43 numbers

$$\left\{\sqrt{2}, \sqrt{3}, \sqrt{5}, \sqrt{6},\ \ldots\ , \sqrt{48}, \sqrt{50}\right\}$$

whose sum has a <u>fraction</u> <u>part</u> least exceeding

$$.51790\ 03017\ 60392\ 24702;$$

unless the integer part of the sum is unusually large or small, we will be able to adjust it by adding a suitable subset of $\{1,2,\ldots,7\}$, bringing the integer part up to 119.

We now have 2^{43} possibilities to try; but <u>they</u> aren't all distinct, either. The values

$$\left\{\sqrt{2}, \sqrt{8}, \sqrt{18}, \sqrt{32}, \sqrt{50}\right\} = \left\{\sqrt{2},\ 2\sqrt{2},\ 3\sqrt{2},\ 4\sqrt{2},\ 5\sqrt{2}\right\}$$

lead to subsets whose sum is restricted to sixteen values

$$0, \sqrt{2},\ \ 2\sqrt{2},\ \ldots,\ 15\sqrt{2},$$

so only 16 of the 32 subsets are different. Similarly, there are only eleven essentially different subsets of

$$\sqrt{3}, \sqrt{12}, \sqrt{27}, \sqrt{48}$$

to try. By means of these observations our program can do in 10 seconds what it would take $(10\cdot 32\cdot 16)/(16\cdot 11) \approx 29$ seconds to do otherwise.

But there are still more than $2^{41} > 10^{12}$ possibilities remaining, and this is far too large; if it takes $100\mu s$ for me to test one subset, I will have time to test only 10^5 subsets.

One way to gain speed is to divide

$$\left\{\sqrt{2}, \sqrt{3},\ \ldots\ , \sqrt{48}, \sqrt{50}\right\}$$

into two parts A and B, and to form a table containing all sums of the subsets of A. Then for each subset of B we can look in the table to see if there is an entry with the appropriate leading bits.

That is what I eventually did. Let

$$A = \left\{\sqrt{2}, \sqrt{8}, \sqrt{18}, \sqrt{32}, \sqrt{50}, \sqrt{3}, \sqrt{12}, \sqrt{27},\right.$$
$$\left.\sqrt{48}, \sqrt{5}, \sqrt{6}, \sqrt{7}, \sqrt{10}, \sqrt{11}, \sqrt{13}, \sqrt{14}, \sqrt{15}\right\},$$

so that the subsets of A take on $(11/16)\cdot 2^{16}$ different values. I stored the 32 leading bits of these fraction parts into a hash table of size 2^{16}, and in another table of size 2^{16} I stored a bit pattern identifying the corresponding subset. (The hash code was the low order 16 bits of each 32 bit key.) Let $B = \{\sqrt{17}, \sqrt{19}, \sqrt{20}, \ldots, \sqrt{47}\}$ be the remaining set of 26 elements; in the machine I used only the 32 leading bits of the fraction parts of each number $\sqrt{k}$. Since my hash table contained $(11/16)\cdot 2^{16}$ more or less random 32-bit numbers, I expected to get a match once out of every $2^{32}/(11/16)\cdot 2^{16} = (16/11)\cdot 2^{16}$ times I looked up another 32-bit value. Therefore I wanted to have a program that generated 2^{17} or so subsets S of B, looking up the 32 bits corresponding to $(.51790 \ldots - \Sigma S) \bmod 1$ in the hash table. I figured that would probably give me two or three matches, and I could choose the best corresponding partition. But I would have only about 75 to 80 μs to spend per subset, so I needed a fast way to probe the hash table and to generate the subset sums.

The solution was to use ordered hash tables with linear probing; this is a variant of hashing which Ole Amble and I had discovered a few years ago [1]. Such hash tables are especially well suited to cases when searches are unsuccessful, requiring only 2.1 probes per search in this case. For the subset generation, I used Gray code [2], since this meant that only one subset element changed state each time and the subset sum was therefore easy to update. It took about 1.5 seconds to build the hash table. I started the subset generation of B in a random part of its cycle, since I knew that I would only be able to look at a small fraction of its subsets.

The best of the three results I got before 10 seconds expired was actually typed out after only 6 seconds, namely

$$\sqrt{2} + \sqrt{3} + \sqrt{4} + \sqrt{5} + \sqrt{6} + \sqrt{12} + \sqrt{17} + \sqrt{19} + \sqrt{22} + \sqrt{27} +$$
$$\sqrt{28} + \sqrt{29} + \sqrt{33} + \sqrt{34} + \sqrt{35} + \sqrt{37} + \sqrt{38} + \sqrt{49} + \sqrt{50}$$
$$\approx 119.51790\ 03021\ 65123\ 39726\ 54768 \quad .$$

This sum differs from its complementary sum by approximately $8\cdot 10^{-10}$. (The optimum partition might well be a thousand times better, as mentioned above; my program could have found it if 10 thousand seconds were available instead of ten!)

I used a fairly large and fast computer, the DEC KL-10 at Stanford's Artificial Intelligence Laboratory. If I had used a slower machine, I would have cut down the number of bits in each hash table entry from 32 to something less, one bit per factor of two in speed. If I had used a smaller machine, I would have had to make the hash table smaller.

If I really wanted to get the optimum partition, I think it could be found in about 30 minutes. The best approach I can think of would be to divide the 43 irrational numbers into two subsets having respectively $11/16 \cdot 2^{21}$ and 2^{21} distinct sums. I would sort each of these files of sums by their fraction parts; since each file contains about two million entries, this would be the time-consuming part of the operation. (Instead of using a general-purpose sort routine I would write a special one, since it is easy to go from the sorted file for set A to the twice-as-long sorted file for $A \cup \{a\}$ by essentially merging the first file with itself. The total time to build the sorted set for A when A has n elements is therefore of order 2^n, while a general-purpose sort routine would take order $n \cdot 2^n$ steps.) Finally, given two large sorted files

$$x_1 \le \ldots \le x_n \quad \text{and} \quad y_1 \le \ldots \le y_n$$

and a number z, there is a nice algorithm which computes

$$\min\{x_i + y_j \mid x_i + y_j \ge z\}.$$

The reader will enjoy discovering this algorithm for himself; curiously it is essentially the same as Hamming's "p/q" algorithm in PC41-4, under a logarithmic transformation, with $x_p = \log p$ and $y_q = \log(qx)$.

I tested my hashing program by trying it first on

$$\{\sqrt{1}, \sqrt{2}, \ \ldots\ , \sqrt{30}\}$$

and then on $\{\sqrt{1}, \sqrt{2}, \ \ldots\ , \sqrt{40}\}$.

I gave the optimum partition for the former case in the previous article (see issue No. 46); and I think I found the optimum partition also in the latter case:

$$\sqrt{2} + \sqrt{5} + \sqrt{7} + \sqrt{13} + \sqrt{14} + \sqrt{17} + \sqrt{18} + \sqrt{19} + \sqrt{22} + \sqrt{23} +$$
$$\sqrt{24} + \sqrt{26} + \sqrt{27} + \sqrt{29} + \sqrt{30} + \sqrt{32} + \sqrt{34} + \sqrt{38} + \sqrt{39}$$
$$\approx 85.807894002346 \ ,$$

$$\sqrt{1} + \sqrt{3} + \sqrt{4} + \sqrt{6} + \sqrt{8} + \sqrt{9} + \ldots + \sqrt{40} \approx 85.807894002154 \ .$$

However, I am not absolutely sure that this is the best, because there is a slight chance that an unusual rounding error might have occurred somewhere in the calculations.

I still think that the 10-second-limited problem is more interesting; according to Hamming's famous aphorism, the purpose of computing is insight, and I believe this problem brings some valuable insights into view.

Postscript: R. L. Graham recently told me about the following sets of nine square roots whose sums agree to 54 decimal places:

$$\sqrt{100000001} + \sqrt{100000025} + \sqrt{100000031} + \sqrt{100000084} + \sqrt{100000087} + \sqrt{100000134} + \sqrt{100000158} + \sqrt{100000182} + \sqrt{100000198}$$

$$= 9000.00449\ 99983\ 56751\ 33987\ 35593\ 13014\ 12703\ 30519\ 82221\ 56985\ 62024\ 86633\ \ldots$$

$$\sqrt{100000002} + \sqrt{100000018} + \sqrt{100000042} + \sqrt{100000066} + \sqrt{100000113} + \sqrt{100000116} + \sqrt{100000169} + \sqrt{100000175} + \sqrt{100000199}$$

$$= 9000.00449\ 99983\ 56751\ 33987\ 35593\ 13014\ 12703\ 30519\ 82221\ 56985\ 62028\ 18260\ \ldots$$

To find the secret of his construction, see Hardy and Wright's <u>Theory of Numbers</u>, 4th edition, p. 338 (notes on Section 21.10).

References

[1] Ole Amble and Donald E. Knuth, "Ordered hash tables," <u>The Computer Journal</u> 17 (May, 1974), 135-142.

[2] Martin Gardner, "Mathematical Games," <u>Scientific American</u> 227 (August, 1972), 106-109.

□

2-3-5 again

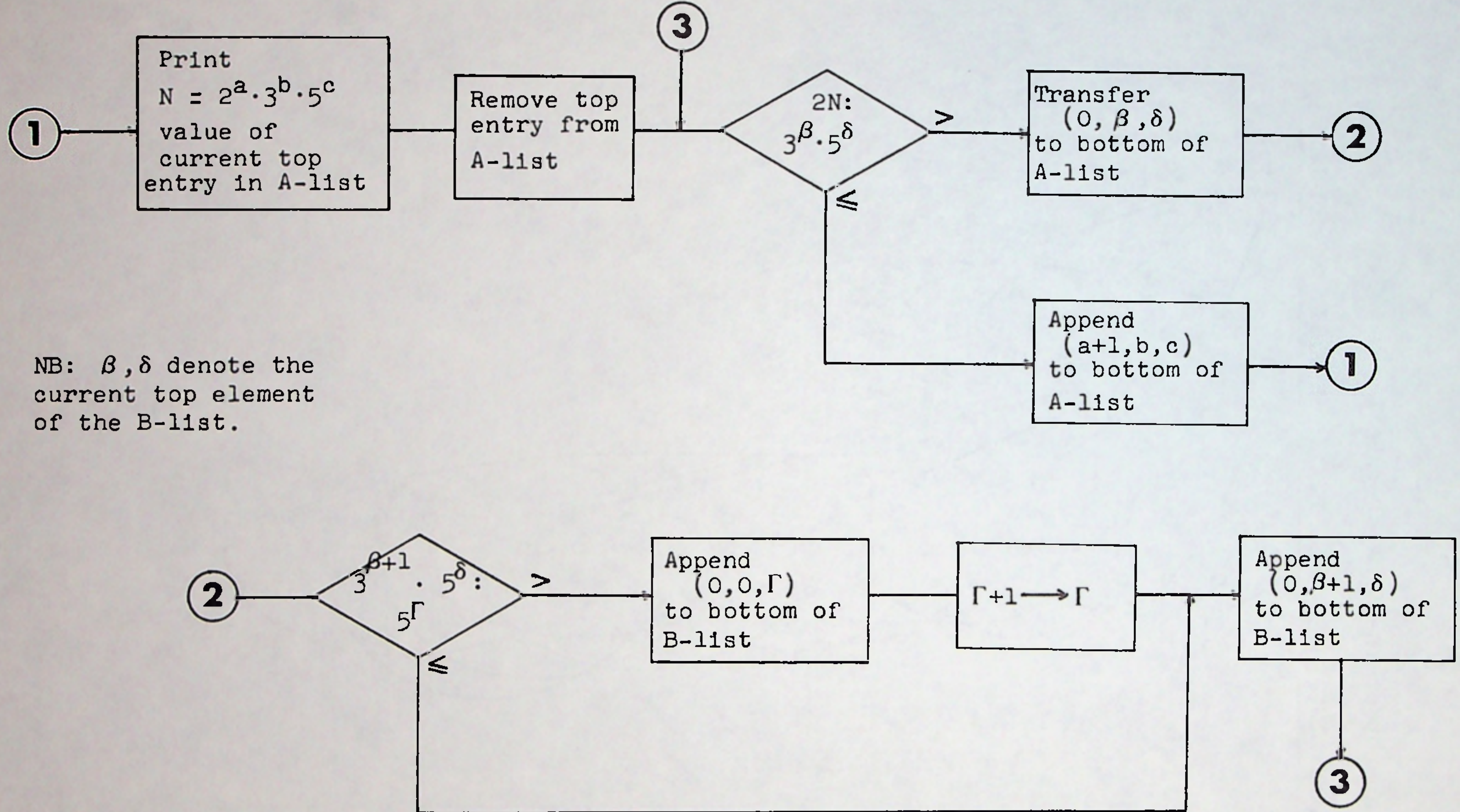
In issue No. 10 there first appeared this problem:

> Show the logic of generating, in order, all numbers having only the factors 2, 3, and/or 5. The first 30 such numbers are:
>
> 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25, 27, 30, 32, 36, 40, 45, 48, 50, 54, 60, 64, 72, 75, 80, 81.

The problem was attributed to Richard Hamming.

In issue No. 16, three different solutions to the 2-3-5 problem were shown, including one by Hamming. The following comes from Norman Sanders, Trondheim, Norway.

> I had the problem of generating thousands of numbers of the form
>
> $$2^a \cdot 3^b, \quad 2^a \cdot 3^b \cdot 5^c, \quad 2^a \cdot 3^b \cdot 5^c \cdot 7^d$$
>
> in connection with the numerical validation of an approximation in prime-number theory. But that was in the mid-50's on EDSAC-1, which had a 1K store, no tapes, no divide, 1.5 millisecond cycle time, and zero-generation reliability. A divide subroutine was a time-consuming luxury that ruled out the first two solutions shown in PC16. The severe storage limitations ruled out long lists, and the low availability of the computer implied a simple program.
>
> As an example of the general method, the solution for
>
> $$2^a \cdot 3^b \cdot 5^c$$
>
> was to generate three lists, A, B, and C (of which C consisted of a single entry only) and to transfer elements from list to list, freeing space at each transfer and each print instruction. In this way, I was able to exceed 2^{100} without reaching the limitation of the store.



NB: β, δ denote the current top element of the B-list.

Norman Sanders' algorithm

for the 2-3-5 problem.

Snapshot at the time of printing N = 25:

PRINT LIST	A-LIST	B-LIST	C-LIST
2	1 0 0	0 1 0	0 0 1
3	0 1 0	0 0 1	0 0 2
4	2 0 0	0 2 0	0 0 3
5	0 0 1	0 1 1	0 0 4
6	1 1 0	0 0 2	
8	3 0 0	0 3 0	
9	0 2 0	0 2 1	
10	1 0 1	0 1 2	
12	2 1 0	0 4 0	
15	0 1 1	0 0 3	
16	4 0 0	0 3 1	
18	1 2 0		
20	2 0 1		
24	3 1 0		
25	0 0 2		
	0 3 0		
	1 1 1		
	5 0 0		
	2 2 0		
	3 0 1		
	0 2 1		
	4 1 0		

The A-list consists of triples (a, b, c), whose values are $2^a \cdot 3^b \cdot 5^c$, in increasing order of value. The B-list consists of triples $(0, \beta, \delta)$, whose values are $3^\beta \cdot 5^\delta$, in order. The C-list contains the single triple $(0,0,\Gamma)$, of value 5^Γ. Initially the A-list consists of the single entry (1, 0, 0); the B-list of the single entry (0, 1, 0), and C of the entry (0, 0, 1). The logic of the algorithm is shown in the flowchart.

The entries in the lists above the dotted lines have been removed. N = 25(0,0,2) has just been printed, and the next step is to compare (1,0,2) (= 50) with (0,1,2) (= 75) at the top of the B-list. It is less, so (1,0,2) will be appended below (4,1,0) in the A-list.

A few pertinent remarks:

> Don't throw anything away. I'm far too experienced today to reproduce such a simple method.
>
> Document your stuff so that you can understand it twenty years later. You never know what the Hamming birds are going to come up with.
>
> When we had hardly any computers, the algorithms we used had to treat the idiosyncracies of the hardware very carefully. Now that we have thousands of computers to choose from, most programming seems to be machine-independent. Computing has become Aristotelian.
>
> We managed to get a lot of useful work out of the primitive computers. If we hadn't they would have stayed primitive, or died out all together. □

[Mr. Sanders is the author of <u>The Corporate Computer; How to Live With an Ecological Intrusion</u>, McGraw-Hill, 1973. His pixie-ish lack of respect for authority should not detract from his computing wisdom.]

In Martin Gardner's "Mathematical Games" column in the November 1976 Scientific American, he shows that the following trick works:

Ask someone to pick a number, write it down, and show it to you. You add one to his number and divide by some other number that you have chosen (and secretly written down) and give him the result. Your friend now has two numbers and is to follow this plan: Given two numbers, add one to the second number, divide by the first number, and record the result as the third number. Then, using the last two of these, repeat the procedure two more times; the final result will be your hidden number.

The arithmetic for the choice of 17 by the victim and 1.2345679 as your hidden number is shown here:

1. Your hidden number: 1.2345679
2. His first number: 17
3. You give him $\dfrac{18}{1.2345679} = 14.58$
4. $\dfrac{15.58}{17} = .9164705882$
5. $\dfrac{1.9164705882}{14.58} = .1314451707$
6. $\dfrac{1.1314451707}{.9164705882} = 1.2345679$

It is easy to show that this must work. The successive stages for any two starting numbers will be:

(1) x	(2) y
(3) $\dfrac{y+1}{x}$	(4) $\dfrac{y+x+1}{xy}$
(5) $\dfrac{x+1}{y}$	(6) x

Thus, the numbers will repeat on a cycle of 5.

Problem: To find a simple similar algorithm that will repeat on a cycle of 6.

Problem Solution

Wendy's Problem (Number 148, PC45-6) called for counting all possible triangles that can be formed on a 10 x 10 grid of points and tabulate the results by areas.

Associate Editor David Babcock produced the tabulation shown here. The column labelled N is twice the area and column K shows how many such triangles there are. The total of the K values is 161,700.

It is of interest to note that there are 4448 combinations of 3 points that do not form a triangle, and the curious distribution of the other 80 possibilities.

N	K	N	K	N	K	N	K	N	K
0	4448	1	5852	2	8376	3	7236	4	9064
5	5816	6	9464	7	4940	8	7812	9	5404
10	6128	11	3160	12	7904	13	2828	14	4760
15	4436	16	5020	17	2260	18	5032	19	2016
20	4160	21	3136	22	2416	23	1600	24	4368
25	1924	26	1888	27	1916	28	2640	29	1112
30	2640	31	976	32	1972	33	1116	34	1120
35	1420	36	1960	37	664	38	840	39	736
40	1300	41	504	42	1336	43	432	44	624
45	692	46	448	47	320	48	932	49	500
50	320	51	296	52	304	53	184	54	480
55	152	56	524	57	176	58	144	59	104
60	168	61	88	62	88	63	260	64	192
65	44	66	72	67	40	68	48	69	48
70	32	71	24	72	148	73	16	74	8
75	16	76	8	77	12	78	8	79	8
80	4	81	36						

We figure to please you.

F-N SEQUENCES

In the sequences shown in Table A, F-1 is the familiar Fibonacci sequence, in which

$$F_n = F_{n-1} + F_{n-2}$$

It is well known that the ratio of one term to the preceding term

$$F_k/F_{k-1}$$

converges to

$$\frac{1+\sqrt{5}}{2} = 1.618033988\ldots$$

The sequence F-2, starting with three ones, is formed by the relation

$$F_n = F_{n-2} + F_{n-3}$$

and it appears to converge to the ratio 1.3247...
We say "appears to converge" because the ratio of one term to the preceding term has a cycle of two values. For example, successive ratios for the first few terms are:

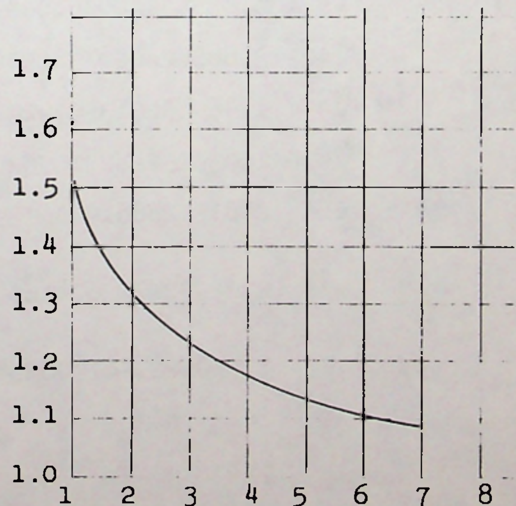
$$16/12 = 1.3333$$
$$21/16 = 1.3125$$
$$28/21 = 1.3333$$
$$37/28 = 1.3214$$
$$49/37 = 1.3243$$
$$65/49 = 1.3265$$
$$86/65 = 1.3231$$
$$114/86 = 1.3256$$
$$151/114 = 1.3246$$
$$200/151 = 1.3245$$

so that the value to which the ratio converges is the average of two ratios. Similarly, for F-3, the convergent value will be found by averaging three successive ratios.

F-1	F-2	F-3	F-4	F-5
1	1	1	1	1
1	1	1	1	1
2	1	1	1	1
3	2	1	1	1
5	2	2	1	1
8	3	2	2	1
13	4	2	2	2
21	5	3	2	2
34	7	4	2	2
55	9	4	3	2
89	12	5	4	2
144	16	7	4	3
233	21	8	4	4
377	28	9	5	4
610	37	12	7	4
987	49	15	8	4

Table A

The first 16 terms of the first five F-N sequences.



Subsequent members of the F-N family of sequences are defined similarly:

F-3:	$F_n = F_{n-3} + F_{n-4}$	converging to 1.2207...
F-4:	$F_n = F_{n-4} + F_{n-5}$	converging to 1.1672...
F-5:	$F_n = F_{n-5} + F_{n-6}$	converging to 1.1347...
F-6:	$F_n = F_{n-6} + F_{n-7}$	converging to 1.1128...

For the sequences F-1 through F-8, the convergent values plot as shown. What is the curve that best expresses the relationship shown in the curve? Is the curve asymptotic to y = 1?

Log 47	1.672097857935717464414219399449200640159803098429948
ln 47	3.850147601710058586820950669772173708896050502020224
$\sqrt{47}$	6.855654600401044124935871449084848960460643461001326
$\sqrt[3]{47}$	3.608826080138694689252517295958892614905551690162338
$\sqrt[10]{47}$	1.469636013359929253623112224371627488588820827562794
$\sqrt[100]{47}$	1.039252262310962953384796629222198650121736878483157
e^{47}	258131288619006739623.28580021527338043163708299304406081061397243682275040978221987
π^{47}	232297222236041657886385.79948902814405844698449614631767505405499833821420635585
$\tan^{-1} 47$	1.549522940770835415559197706584603144431776847570469

Cunningham's Process

In the May, 1962 American Mathematical MONTHLY, G. S. Cunningham showed that any positive fraction can be represented as a sum of distinct unit fractions.

$$\frac{a}{b} = \underbrace{\frac{1}{b} + \frac{1}{b} + \frac{1}{b} + \cdots + \frac{1}{b}}_{\text{a terms}}$$

Retain one of the duplicating terms. Replace each of the others by the identity

$$\frac{1}{n} = \frac{1}{n+1} + \frac{1}{(n+1)n}$$

and repeat this process until all terms are distinct.

For example:

$$\frac{2}{3} = \frac{1}{3} + \frac{1}{3}$$

$$= \frac{1}{3} + \frac{1}{4} + \frac{1}{12}$$

$$\frac{3}{4} = \frac{1}{4} + \frac{1}{4} + \frac{1}{4}$$

$$= \frac{1}{4} + \frac{1}{5} + \frac{1}{20} + \frac{1}{5} + \frac{1}{20}$$

$$= \frac{1}{4} + \frac{1}{5} + \frac{1}{20} + \frac{1}{6} + \frac{1}{30} + \frac{1}{21} + \frac{1}{420}$$

$$\frac{4}{5} = \frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{5}$$

$$= \frac{1}{5} + \frac{1}{6} + \frac{1}{30} + \frac{1}{6} + \frac{1}{30} + \frac{1}{6} + \frac{1}{30}$$

$$= \frac{1}{5} + \frac{1}{6} + \frac{1}{30} + \frac{1}{7} + \frac{1}{42} + \frac{1}{31} + \frac{1}{930} + \frac{1}{7} + \frac{1}{42} + \frac{1}{31} + \frac{1}{930}$$

$$= \frac{1}{5} + \frac{1}{6} + \frac{1}{30} + \frac{1}{7} + \frac{1}{42} + \frac{1}{31} + \frac{1}{930} + \frac{1}{8} + \frac{1}{56} + \frac{1}{43} + \frac{1}{43\cdot 42}$$
$$+ \frac{1}{32} + \frac{1}{32\cdot 31} + \frac{1}{931} + \frac{1}{931\cdot 930}$$

$$= \frac{1}{5} + \frac{1}{6} + \frac{1}{30} + \frac{1}{7} + \frac{1}{42} + \frac{1}{31} + \frac{1}{930} + \frac{1}{8} + \frac{1}{56} + \frac{1}{43} + \frac{1}{1806}$$
$$+ \frac{1}{32} + \frac{1}{992} + \frac{1}{931} + \frac{1}{865830}$$

In each example, the fraction used was $\frac{N-1}{N}$

Problem: extend this table:

N	Number of final terms	Largest denominator
3	3	12
4	7	420
5	15	865830

ANNUAL INDEX

Volume 4 1976
Issues 34 through 45
230 pages